

# Как с помощью BPMS (jBPM) заместить продукты SAS?

Олег Терёшкин

СБЕР



# Обо мне



Олег Терёшкин

IT-лидер кластера

## Опыт

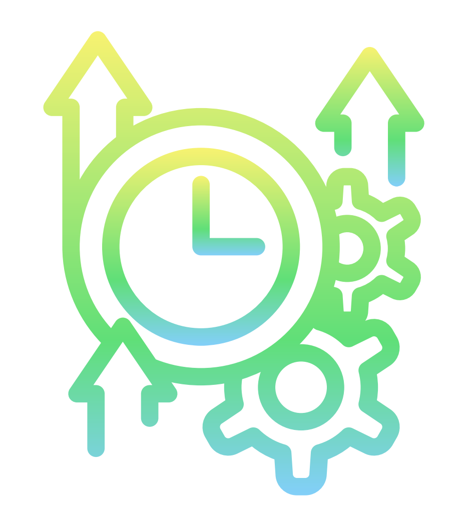
- Низкоуровневая разработка (драйверы)
- Data Engineering
- Разработка микросервисов
- Solution-архитектура

## Образование

Мехмат МГУ



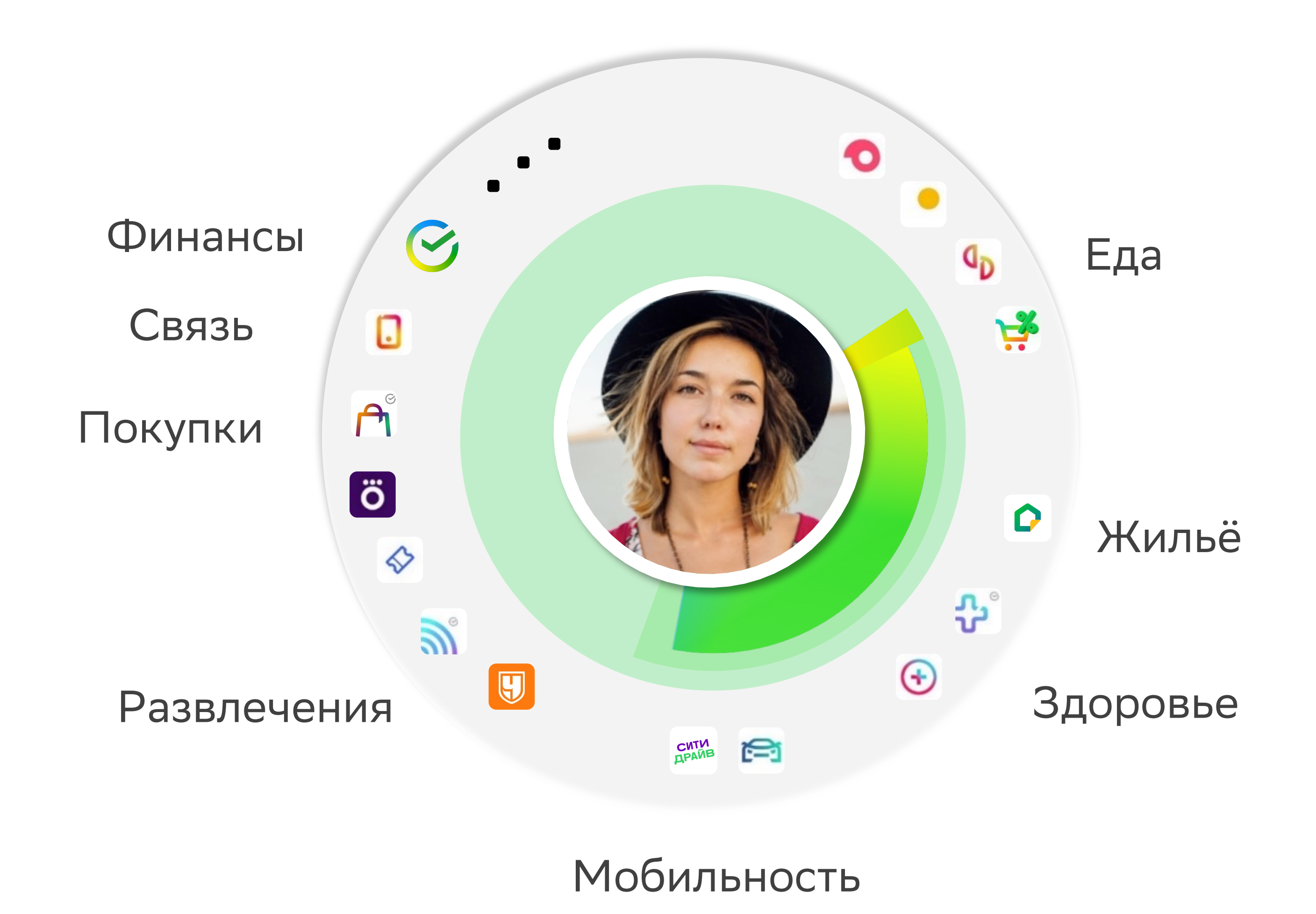

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ М.В. ЛОМОНОСОВА



# План доклада



# Платформа персонализированных предложений для клиентов СБЕРа




**1+ млрд**

персональных предложений в месяц

**106 млн**

клиентов

**10+ тыс.**

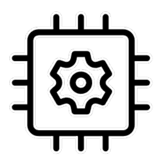
рекламных кампаний в месяц

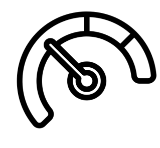
**90+**

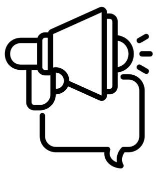
компаний-партнёров

# Выручка SAS за 2021 — 3,5 млрд рублей

## Основные направления деятельности


Платформа исполнения моделей


Скоринг


Автоматизация маркетинговых кампаний

## Основные заказчики









## Наиболее известные продукты







MA

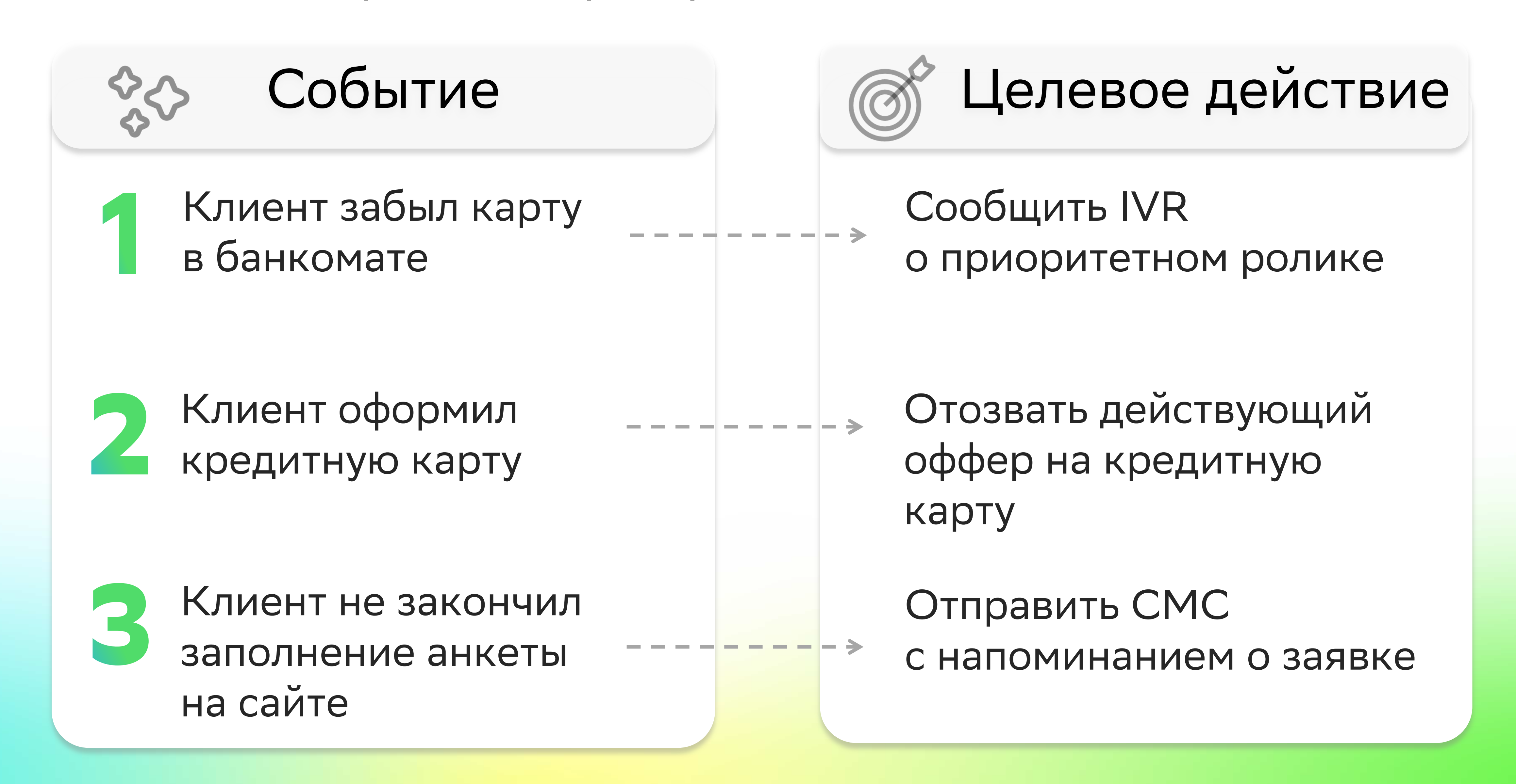
RTDM

MO

EG

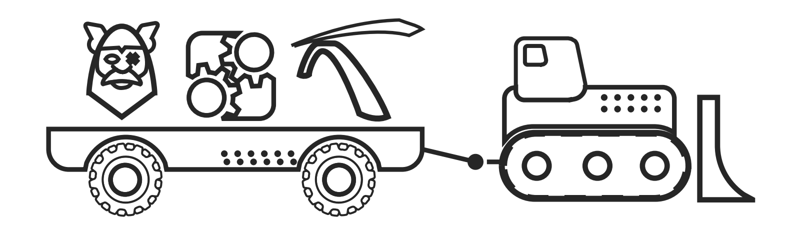
...

# Типовые сценарии, которые решали с помощью SAS



# SAS ушёл с российского рынка. Начинаем задачу миграции





Компания SAS покинула российский рынок в марте 2022

СБЕР остался без экспертной поддержки внедренных решений (RTDM, Viya, DI, EG, ...)

Уход от vendor lock-in (SAS2NULL)

Экспертная команда разработки от SAS прекратила свое существование

Срок окончания лицензии по SAS RTDM – ноябрь 2022

# Основные требования к движку

**1** Open-source (Java Stack)

**2** Low-code/no-code для настройки бизнес-правил (DMN, Drools, Data objects)

**3** Highload: горизонтально масштабируемое решение под нагрузки ~50 000 TPS

**4** Интеграции из коробки: Kafka, REST API

**20+ решений на рынке**

Вендоры:







Open-source:










...

# Продукт глазами пользователя



# Артефакты, которые исполняет jBPM

**Результат билда – 2 артефакта**

Demo.pom
Demo.jar

M2 Repository Content

Upload

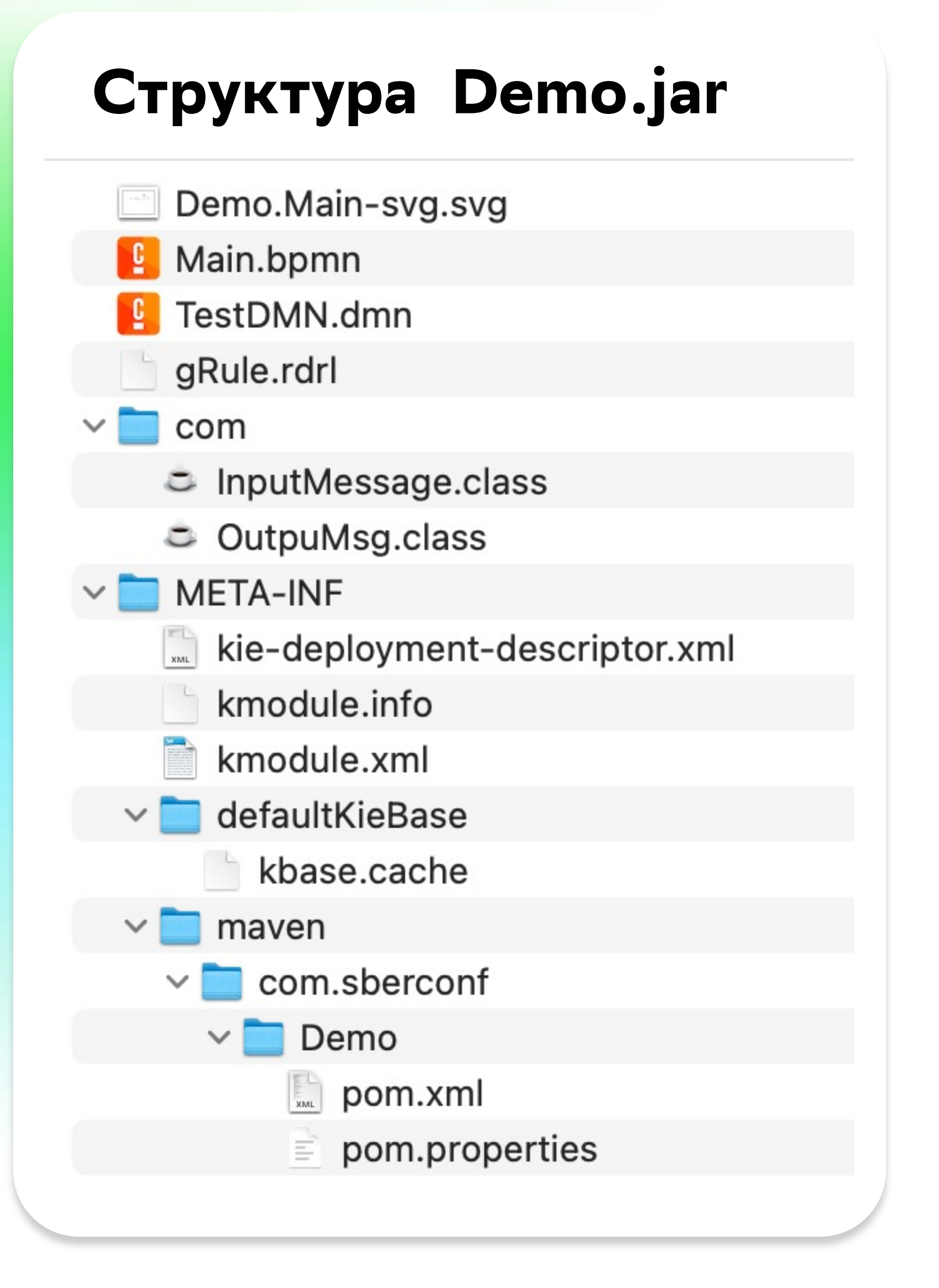
| Name | GAV | Last Modified | Open | Download |
|---|---|---|---|---|
| Demo-1.0.0-20221116.144028-16.pom | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 16 17:40:28 | Open | Download |
| Demo-1.0.0-20221116.144028-16.jar | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 16 17:40:28 | Open | Download |
| Demo-1.0.0-20221116.143837-15.jar | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 16 17:38:37 | Open | Download |
| Demo-1.0.0-20221116.143837-15.pom | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 16 17:38:37 | Open | Download |
| Demo-1.0.0-20221109.173010-14.jar | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 9 20:30:10 | Open | Download |
| Demo-1.0.0-20221109.173010-14.pom | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 9 20:30:10 | Open | Download |
| Demo-1.0.0-20221109.172622-13.jar | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 9 20:26:23 | Open | Download |
| Demo-1.0.0-20221109.172622-13.pom | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 9 20:26:23 | Open | Download |
| Demo-1.0.0-20221109.172259-12.jar | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 9 20:22:59 | Open | Download |
| Demo-1.0.0-20221109.172259-12.pom | com.sberconf:Demo:1.0.0-SNAPSHOT | 2022 Nov 9 20:22:59 | Open | Download |

# Структура JAR

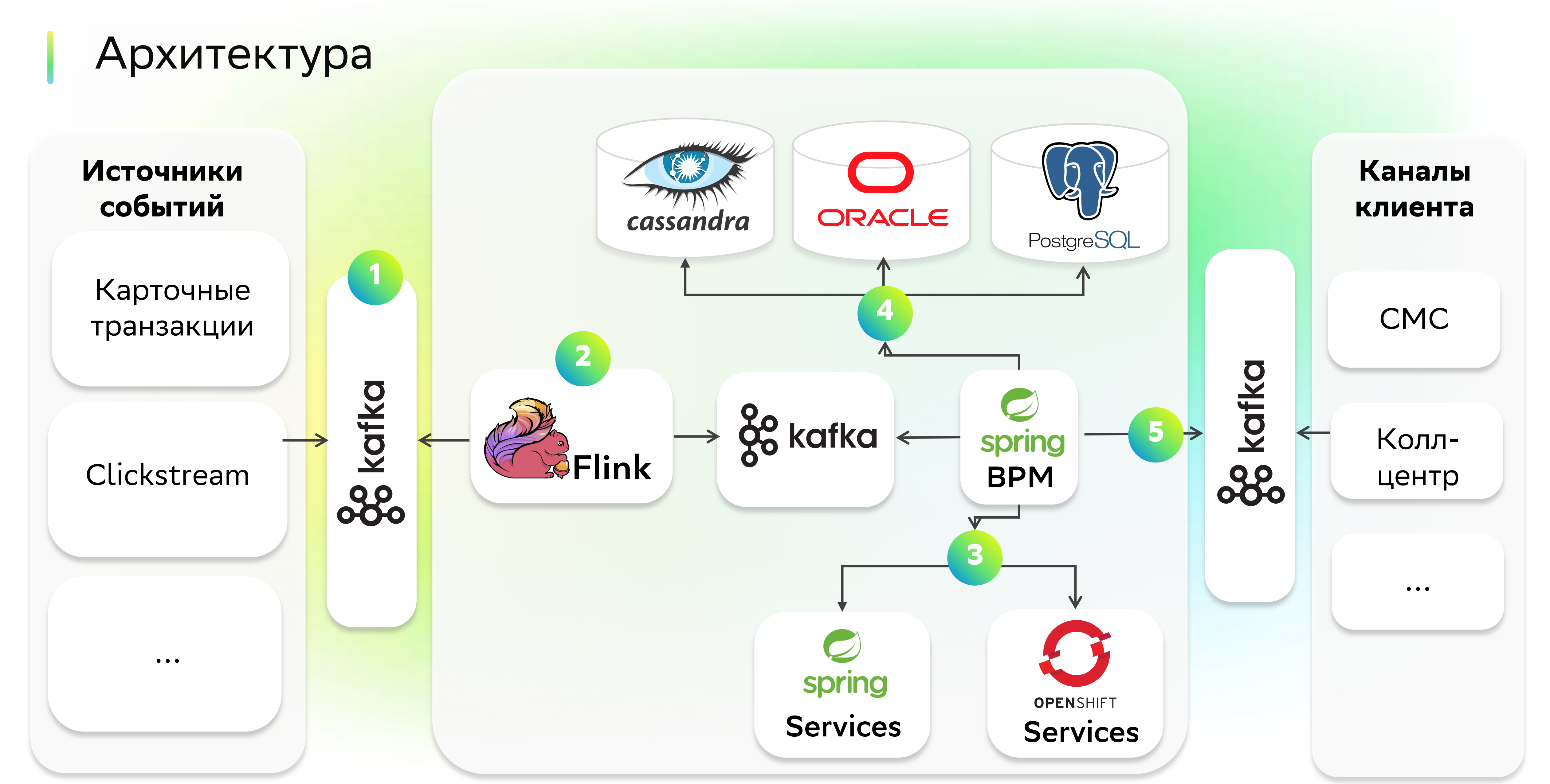
XML-схема процесса и low-code ------>

наши классы входящего
и исходящего сообщений ------>

конфигурации исполнения
процесса (persistence, audit) ------>



# Архитектура



# Характеристики железа для нагрузочного тестирования

| | Кол-во серверов | ЦПУ, кол-во ядер | ОЗУ, ГБ | Диск, ГБ |
|---|---|---|---|---|
| kafka | 5 | 4 | 16 | 500 |
| jBPM | 11 | 4 | 16 | 100 |
| Flink | 16 | 4 | 32 | 1024 |

# jBPM — методика проведения нагрузочного тестирования

**Поиск максимума:**

**1 ч. 10 мин.**
длительность теста

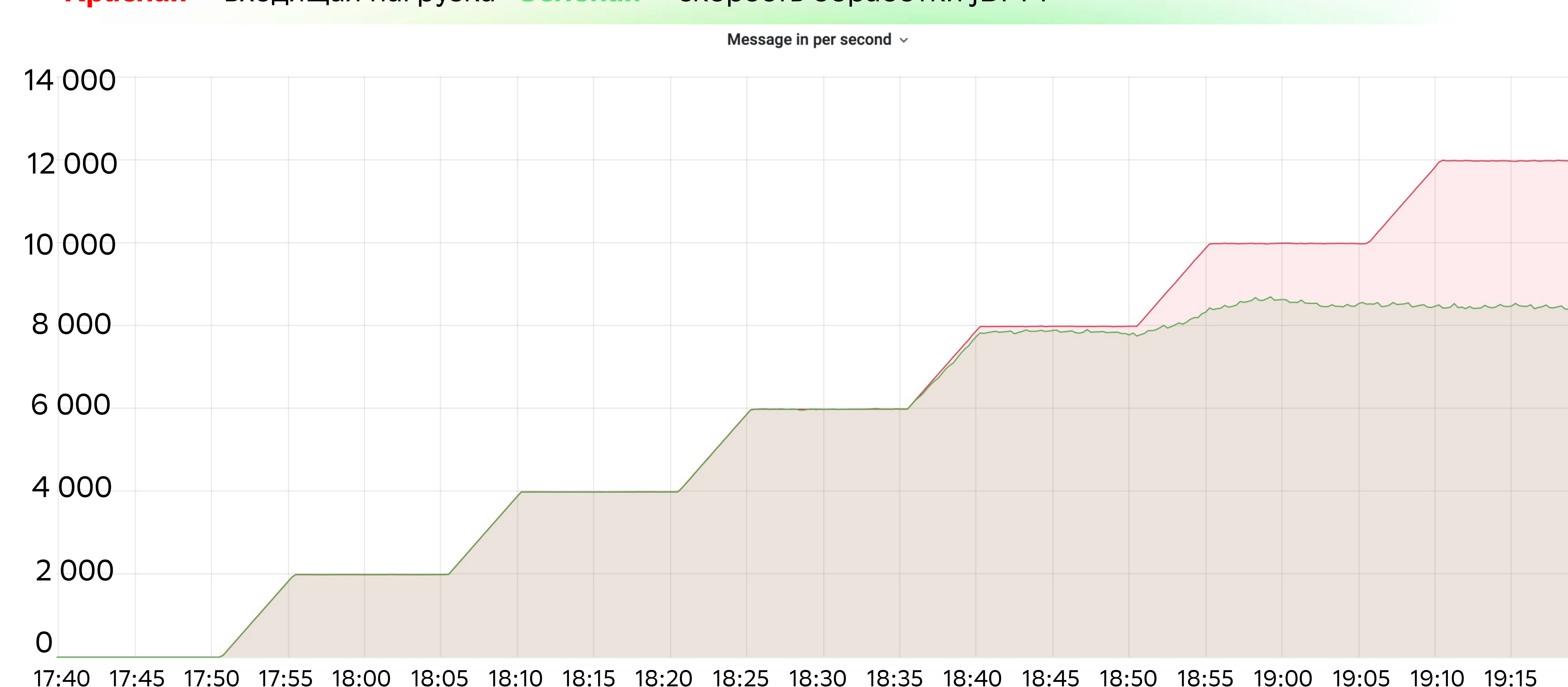
**2 000 TPS**
начальный уровень нагрузки на систему

**2 000 TPS**
шаг каждые 15 минут

**12 000 TPS**
уровень нагрузки на последней ступени теста

# 6000 TPS — максимальная зафиксированная производительность

**Красная** – входящая нагрузка **Зелёная** – скорость обработки jBPM



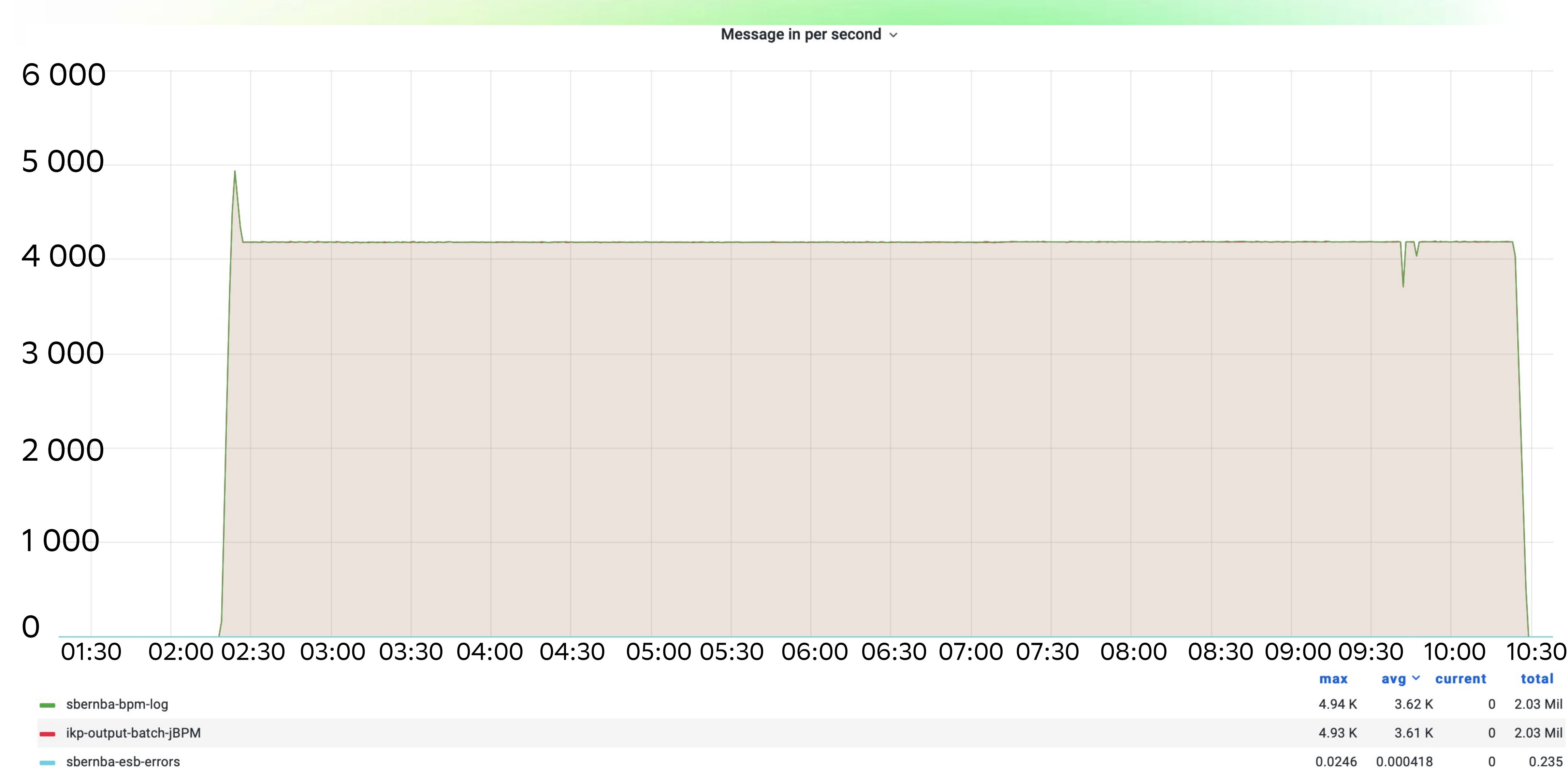
# jBPM — методика проведения нагрузочного тестирования

**Тест надёжности:**

| **8 ч.** | **120 млн** | **4 200 TPS** |
|---|---|---|
| длительность теста | общее кол-во сообщений | надежность системы на уровне производительности подтверждена |

# 4200 TPS — тест надежности

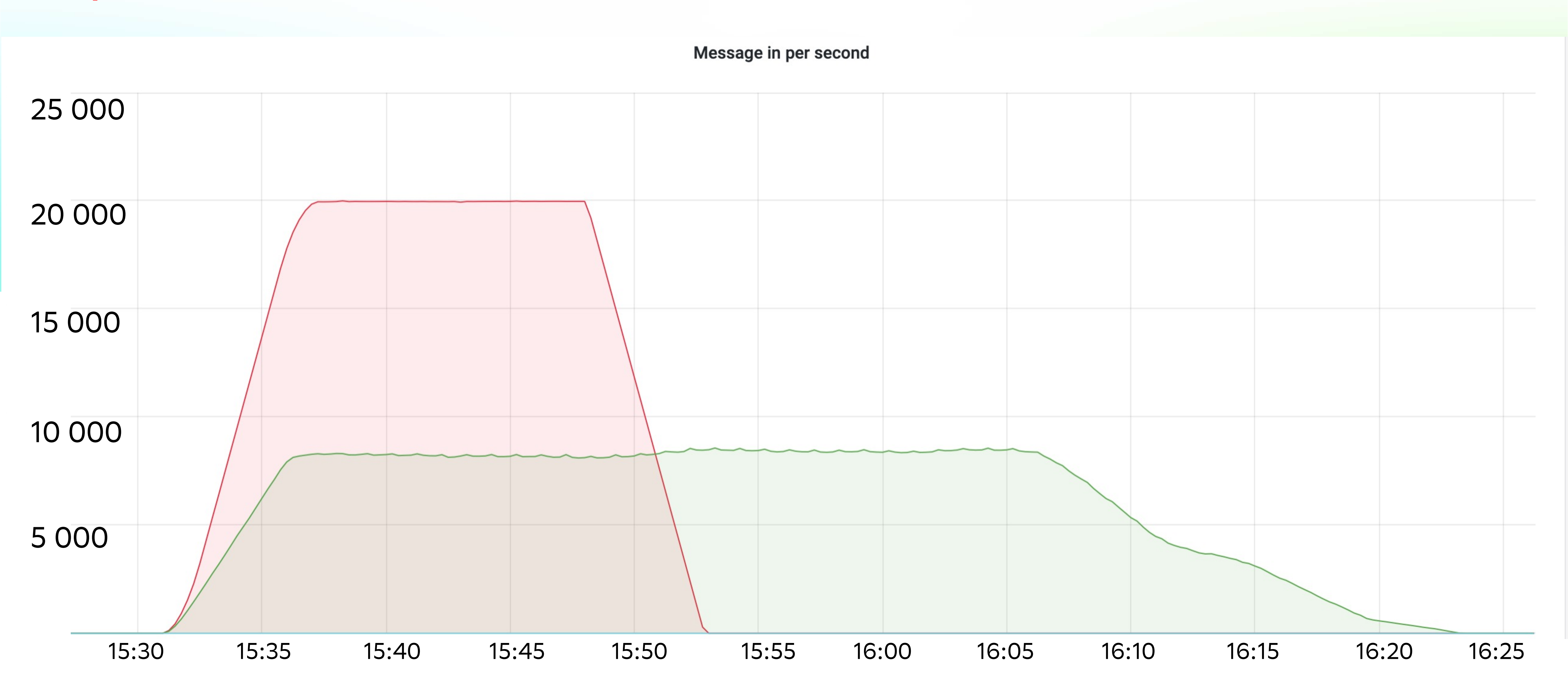
**Красная** – входящая нагрузка **Зелёная** – скорость обработки jBPM




| | max | avg | current | total |
|---|---|---|---|---|
| sbernba-bpm-log | 4.94 K | 3.62 K | 0 | 2.03 Mil |
| ikp-output-batch-jBPM | 4.93 K | 3.61 K | 0 | 2.03 Mil |
| sbernba-esb-errors | 0.0246 | 0.000418 | 0 | 0.235 |

# Потеря сообщений в случае накопления лага

**Красная** – входящая нагрузка **Зелёная** – скорость обработки jBPM



# Шаги в сторону Highload

1

## Хранить в памяти сервисную базу данных движка

```
# service data source configuration
spring.datasource.username=sa
spring.datasource.password=sa
spring.datasource.url=jdbc:h2:mem:spring-boot-jbpm
spring.datasource.driver-class-name=org.h2.Driver
```

2

## Крон по удалению данных из таблиц — не переполнить RAM и не потерять зависшие процессы

```
clean.db.jpa.enable=true
clean.db.sql.enable=true
clean.db.rate=60000
clean.db.delay=60000
clean.db.sql.requestInfo=DELETE FROM REQUESTINFO where timestamp < systimestamp - interval '5' minute
                         AND (status <> 'QUEUED' AND status <> 'RETRYING' AND status <> 'RUNNING' )
clean.db.sql.errorInfo=DELETE FROM ERRORINFO where timestamp < systimestamp - interval '5' minute
clean.db.jpa.period=1m
```

# Шаги в сторону Highload

3

**Потеря процессов в случае накопления лага**

A Использование **Retry**

B **RetryCount** = 5,10 …

C **RetryDelay** = …

D Использование топика для ошибок и отдельного процесса для зацикливания (только осторожно) 😁

# Шаги в сторону Highload

4

**Исполнение SQL-запросов с использованием bind'ов**

```
statement.executeQuery("SELECT * FROM s_user where userid = " + i);

pstatement = conn.prepareStatement("SELECT * FROM s_user where userid = ?");
pstatement.setInt(1, i);
pstatement.executeQuery();
```

# Вместо заключения

1. BPMS jBPM можно использовать под нагрузками, и не только для human-процессов
2. Выбирайте решение, максимально закрывающее ваши потребности, и дорабатывайте его под себя
3. Задавайте вопросы в открытое комьюнити, с вашей проблемой/ошибкой уже кто-то сталкивался
4. Не пытайтесь перенести весь код на BPMS

# СПАСИБО ЗА ВНИМАНИЕ!



Оценить доклад

Контакты:
Олег Терёшкин
tg: @o_terjeshkin